Advanced Database Systems, second intermediate test – 4 June 2019 – V1.0

Please feel free to answer this test in English, Italian, or any mixture

1. Consider the following log content. Assume that the DB was identical to the buffer before the beginning of this log, and consider a undo-redo protocol

    (begin,T1) (W,T1,A,0,20) (begin,T2) (W,T2,B,0,30) (begin-ckp,{T1,T2})
    (W,T2,B,30,50) (W,T1,C,0,30) (end-ckp) (begin,T3) (commit,T2) (W,T3,B,50,70)
    (begin,T4) (W,T4,D,0,50) (commit,T4)

    a. Before starting this log, what was the content of A, B, C and D in the PS (Persistent Store)?
    b. Assume there was a crash at the end of the logging period. At crash time, what was the content of A, B, C, D in the buffer? What can be said about the content of A, B, C, D in the PS?
    c. At restart time, which transactions are put in the undo-list? Which in the redo-list?
    d. List the operations that are redone, in the order in which are redone
    e. List the operations that are undone, in the order in which are undone
    f. After restart is finished, what is the content of A, B, C, D in the buffer?
    g. Undo and Redo are executed in the buffer or on the PS?
    h. After restart is finished, what is the content of A, B, C, D in the PS? What is different between this answer and that of question (b)?

2. Assume that a system with no scheduler produces the following history, where we omit the commits:

    $r_1[B], w_3[A], r_2[B], w_2[C], w_3[B], r_1[C], r_2[A], w_3[A]$

    a. Is this history c-serializable?
    b. Exhibit a history that may be produced by a strict 2PL scheduler if presented with the above operations in that order, assuming that each transaction commits immediately after its last operation. If a deadlock takes place, assume that the system just stops.

3. Consider the following tables:

    Sales(Date,FKShop,FKCust,FKProd,UnitPrice,Q,TotPrice)
    Shops(PKShop, ShopName, FKCity)
    Cities(PKCity, CityName, CountryName, Region)

With the following sizes:
   Sales: NRec: 10.000.000, Npag: 100.000; Shops: NRec: 1000, Npag: 5; Cities: NRec: 200, Npag: 1

   Assume a buffer of 1.000 pages. Consider the following queries:

        SELECT C.CountryName, C.Region, Sum(S.TotPrice)
        FROM Sales S, Shops Sh, Cities C
        WHERE S.FKShop=Sh.PKShop AND Sh.FKCity = C.PKCity AND S.Date >
        1/1/2018
        GROUP BY C.CountryName, C.Region

        SELECT C.CountryName, C.Region, Year(S.Date), Sum(S.TotPrice)
        FROM Sales S, Shops Sh, Cities C
        WHERE S.FKShop=Sh.PKShop AND Sh.FKCity = C.PKCity
        GROUP BY C.CountryName, C.Region, Year(S.Date)

   a. In the context of a traditional database, choose a physical organization to optimize both queries, without partitioning or denormalization, and give a synthetic justification of your choice. Specify the primary organization of each table and which indices would you use. Assume that the database contains 1000 different dates each with a similar amount of sales, and that the condition S.Date > 1/1/2018 has a selectivity factor of 10%.
   b. Show an access plan for the first query assuming that all tables are stored serially and no index is used, and compute its cost. Assume each field occupies 4 bytes, and pages measure 4000 bytes, if needed.

4. Answer the following questions. Please keep the answers short and WRITE IN YOUR BEST HANDWRITING.

   a. What is the difference between a distributed database and a parallel database
   b. Assume that we have 5 copies of a piece of data. Give three different examples of Read and Write quorums to obtain a consistent access to that piece of data.

CONTINUES ON THE BACK WITH n5

5. Assume a database
Occupations(OccId,FlatId*,CustomerId*,OwnerId*,From,To,Amount),
Flats(FlatId,Street,StreetNo,City,Country), Customers(CustomerId,Name,Surname,
Street,StreetNo,City,Country), Owners(OwnerId,Name,Surname,
Street,StreetNo,City,Country), where the following table describes the sizes of the
relations:

	NRec	NPag
Occupations	40,000,000	250,000
Flats	400,000	10,000
Customers	10,000,000	200,000
Owners	200,000	4,000

Each database fact (each occupation) describes the fact that a customer occupied a
Flat From a given date To a given date and payed a given Amount. OwnerId*
denotes the Owner of the Flat.
A parallel machine with 1000 nodes is used in order to store the database.
Occupations, Customers and Owners are distributed on all nodes, with no
replication, with a round-robin technique. Flats are hashed on FlatId.
Assume that we want to compute the following query:
SELECT total(o.Amount)
FROM Occupations o, Customers c
WHERE o.CustomerId = c.CustomerId
GROUP BY c.Country, o.From

a. Consider the following strategy:
   a. Redistribute Occupations and Customers over all nodes, with no projection,
      hashing both over CutomerId
   b. Locally compute the join
   c. Redistribute the result over all nodes by hashing on c.Country, o.From
   d. Locally compute the Group-By
   Assume that pages have a size of 4K, you need 10 ms to read/write a page, and
1 ms to send or to receive one page over the network. Assume every machine has a
buffer size of 1.000 pages that can be used to perform database operations. Assume a
model where you sum cost of reading + cost of sending + cost of receiving + cost of
writing, ignoring any parallelism between these four phases.
   a. Compute the time that is needed in order to complete the algorithm described. If
      you miss any information, just ask me.
   b. Suggest some possible optimizations to the strategy of point a – just suggest, do
      not compute anything
   c. Suggest a different data distribution strategy in order to optimize this query